Decorator:

Input - Decorator:

app.component.html

```
<h1> Decorator </h1>
<div class = "container mt-4">
  <div class = "row">
    <div class = "col-6">
      <form>
        <div class = "form-group">
          <lable> Parent Component </lable>
          <input type = "text" class = "form-control #ch>
        </div>
        <button type = "button" class = "bth btn-primary"> Send
                                              </button>
      </form>
    </div>
    <div class = "col-6"> <app-child> </app-child> </div>
  </div>
</div>
```

child.component.html

```
<p> child works! </p>
```

Output:-

Decorator

parent Component   child works!

[   ]

[Send]

@ Input decorator is used to receive the data from parent component to child component.

</div>

app.component.html

```html
<h1> Decorator </h1>
  <div class="Container
                container">
    <div class="row">
      <div class="col-6">
        <div class="form-group">
        <lable> Parent Component </lable>
        <input type="text" class="form-control" #ch>
        </div>
        <button class="btn btn-primary" (click)="onClick(ch.value)">
                          Send </button>

        </div>
        <div class="col-6"> <app-child [receive]="send">
                          </app-child></div>
      </div>
  </div>
```

app.component.ts

```typescript
import {component} from '@angular/core';
@Component({
  selector:
  templateUrl:
  styleUrls:
})
export class AppComponent{
  title = "Decorator Online";
  send:string
  onClick(value)
  {
    this.send=value
  }
}
```

child.component.ts

```
export class childComponent implements OnInit {
  @Input() receive;
  constructor () { }
  ngOnInit () { }
}
```

child.component.html

```
<p> child works ! </p>
<h1> {{receive}} </h1>
```

Output:-

Decorator

parent component

| with the help of input decorate sendingdata |

child works ?
with the help of
input decorate sending
data.

@ Output Decorator:- @Output decorator will always
send the data with the help of EventEmitter()

- emit is the method which is present inside
  EventEmitter object.

- msgdata variable is treated as custom event.

app.component.html

```
<div class="now">
  <div class="col-6">
    <h1> {{receive_msg}} </h1>
  </div>
    <div class="col-6">
      <app-child2 (msgdata)="receive_msg=$event">
      </div>                          </app-child2>
       </div>
    </div>
```

app·component·ts

```
export class AppComponent {
  title = "Decorator Online Class";
  send: string
  receive-msg;
  onclick(value) {
    this·send = value
  }
}
```

child2·component·html

```
<p> Child2 works </p>
<input type = "text" class = "form-control" #ch>
<button class = "bth btn-primary" (click) = "onsend(
                              ch·value)"> send </button>
```

child2-component·ts

```
export class child2Component implements OnInit {
  constructor() {}
  @Output() msgdata = new EventEmitter();
  ngOnInit() {
    onsend (value)
    {
      this·msgdata·emit (value)
    }
  }
}
```

Outputs-

parent Component

┌─────────────────────┐
│ Hi   │   Hi
└─────────────────────┘
[Submit]
                                child2 works
        Hello·   ┌─────────────────────┐
                                │ Hello   │
                                └─────────────────────┘
                                [Submit]
```